Matheuristics to stabilize column generation: application to a technician routing problem

Nicolas Dupin, Rémi Parize, El-Ghazali Talbi

Matheuristics to stabilize column generation: application to a technician routing problem

Nicolas Dupin[1], Rémi Parize, El-Ghazali Talbi[1],

Univ. Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

Abstract. This paper considers a simplified Technician Routing and Scheduling Problem with skill constraints, where a Column Generation (CG) heuristic was shown effective. This work proposes new CG schemes to stabilize the CG process and thus to accelerate the CG heuristics. Solving CG subproblems with matheuristics allows to diversify the column generation. Furthermore, a tabu search matheuristic allows to generate aggressively columns at each iteration. Both techniques imply a better stability of the CG scheme: the diversification is interesting for the first iterations whereas tabu intensification is especially useful for the last iterations. It implies a significant acceleration of the CG convergence. A perspective of these CG strategies proposed is an extension to other problems where CG induces independent and heterogeneous subproblems.

1 Introduction

The technician routing and scheduling (TRS) problem was defined for the Challenge ROADEF [8]. The optimization assigns technicians to vehicles to serve customers' requests, with constraints for skills and number of technicians required for the interventions as well as capacity constraints to use tools. Matheuristics were used to tackle this problem, using Mixed Integer Programming (MIP) for subproblems [11, 13, 17]. This paper considers a relaxation, a multiple depot vehicle routing problems with time windows (VRPTW) where skill constraints define the jobs that vehicles can process. This relaxation was already studied in [7], analyzing the impact of the skill constraints in highly constrained instances and providing tailored matheuristic approaches for such instances.

Column Generation (CG, see [15]) approaches are known to be efficient to solve VRPTW to optimality [9]. Facing large instances, a CG scheme can give rise to primal heuristics [3, 19]. A bottleneck of CG algorithms is the difficulty to have a stable convergence, with "bang-bang" effects and erratic dual variables between consecutive iterations. To stabilize the convergence of dual variables and to reduce the computational effort, stabilization techniques are essential. In that goal, mathematical properties can be exploited as in [2, 4, 18], but also heuristics as in [12].

The contribution of this paper is to investigate matheuristic techniques for the stabilization of the CG algorithm in [7], to accelerate both CG computations

of dual bounds and CG heuristics. Similarly to [5], matheuristics are not only techniques using exact methods within a meta-heuristic, but also techniques increasing the size of instances where dual bounds for MIPs are computable.

This paper is organized as follows. In section 2, we introduce the applicative technician routing and scheduling problem and its compact MIP formulation. In section 3, we present the extended MIP formulations of the problem and its standard CG algorithm. In section 4, an alternative CG scheme is proposed to enhance the diversification among generated columns. In section 5, a matheuristic algorithm is proposed to generate intensively columns for CG subproblems. In section 6, the computational results are discussed. In section 7, our contributions are summarized, discussing also future directions of research.

2 Problem statement

In the following simplification of [8], we assume that the jobs are realized by a single technician. In this case, the skill constraints are binary: either a vehicle can process a job, or it cannot. Furthermore, we relax scheduling and capacity constraints from [8], focusing on the Technician Routing problem.

The routing optimization concerns a set of technicians, who have to realize jobs at customers' places with time window constraints on the arrival time. The jobs require specific skills, which limits the possible routes of the vehicles. The optimization horizon is one day. The objective function to minimize is the total distance of the routes, adding high penalties for non allocated jobs. We note that a similar relaxation of [8] with additional constraints was also studied with a matheuristic approach in [16]. Notations are gathered in the Table 1.

Table 1. Table of notations, set and indexes

$\mathcal{I}$	Set of technicians.
$\mathcal{J}$	Set of customers, the jobs for the technicians.
$\mathcal{J}_i$	Subset of jobs that technician i can complete with skill constraints.
D_j	Duration of job j.
P_j	Cost penalization (or outsourcing cost) if job j is not planned.
d_i	Starting and finishing depot for technician i.
$D(j, j')$	Distance between the location of jobs j and j'.
$D(i, j)$	Distance between the depot d_i to the location of job j.
$T(j, j')$	Transportation time between the location of jobs j and j'.
$T(i, j)$	Transportation time between the depot d_i to the location of job j.
$[\tilde{t}_i^{start}, \tilde{t}_i^{end}]$	Working time window for technician i, to leave and come back to the depot.
$[\tilde{t}_j^{min}, \tilde{t}_j^{max}]$	Time windows to begin the job $j \in \mathcal{J}$.

A first MIP formulation can be written similarly to [1]. Binaries $u^i_{j,j'} \in \{0, 1\}$ are defined for all technician $i \in \mathcal{I}$ and all jobs $(j, j') \in \mathcal{J}_i^2$ related to i , where $u^i_{j,j'} = 1$ indicates that technician i realizes job j and then job j'. By convention $u^i_{j,j} = 0$. We extend these notations with binaries for all technicians $i \in \mathcal{I}$ and

all jobs $j \in \mathcal{J}_i$ with $u^i_{d_i,j}$ (and u^i_{j,d_i} respectively) indicating if job j is the first (respectively the last) job of technician i, and $u^{i'}_{d_i,j} = 0$ if $i \neq i'$. Furthermore, continuous variables t_j represent the start time for job j, bounded with the time-windows constraints: for all $j \in \mathcal{J}$, $\tilde{t}^{min}_j \leqslant t_j \leqslant \tilde{t}^{max}_j$.

$$\min_{u^i_{j,j'}, t_j} \sum_{j,j' \in \mathcal{J} \cup \mathcal{I}} D(j,j') u^i_{j,j'} + \sum_{j \in \mathcal{J}_i} P_j \left(1 - \sum_{i,j'} u^i_{j',j}\right) \tag{1}$$

$$\forall j, \quad \sum_{i \in \mathcal{I}} \sum_{j' \in \mathcal{J}_i \cup \{d_i\}} u^i_{j'j} \leqslant 1 \tag{2}$$

$$\forall i \in \mathcal{I}, j, \quad \sum_{j' \in \mathcal{J}_i \cup \{d_i\}} u^i_{j'j} = \sum_{j' \in \mathcal{J}_i \cup \{d_i\}} u^i_{jj'} \tag{3}$$

$$\forall i \in \mathcal{I}, j, \quad \sum_{j' \in \mathcal{J}_i \cup \{d_i\}} u^i_{j'j} \leqslant 1 \tag{4}$$

$$\forall i \in \mathcal{I}, \quad \sum_{j' \in \mathcal{J}_i} u^i_{j'd_i} = \sum_{j' \in \mathcal{J}} u^i_{d_i j'} \leqslant 1 \tag{5}$$

$$\forall (j,j') \in \mathcal{J}^2, \quad t_j + D_j + T(j,j') \leqslant t_{j'} + \left(1 - \sum_i u^i_{j,j'}\right) \cdot M^1_{j,j'} \tag{6}$$

$$\forall i \in \mathcal{I}, j \in \mathcal{J}_i, \quad \tilde{t}^{start}_i + T(i,j) \leqslant t_j + \left(1 - u^i_{d_i j}\right) \cdot M^2_{i,j} \tag{7}$$

$$\forall i \in \mathcal{I}, j \in \mathcal{J}_i, \quad t_j + D_j + T(j,i) \leqslant \tilde{t}^{end}_i + \left(1 - u^i_{j d_i}\right) \cdot M^3_{i,j} \tag{8}$$

$$\forall i \in \mathcal{I}, j, j' \quad u^i_{j,j'} \in \{0,1\}, t_j \in [\tilde{t}^{min}_j, \tilde{t}^{max}_j] \tag{9}$$

The objective (1) to minimize is the distance of the total routing, with penalties for non attributed jobs. Constraints (2) are elementary constraints. Constraints (3) are flow constraints to model tours. Constraints (4) are capacity flow constraints, expressing that one job is done at most one time for an optimal solution. Indeed, having $d(j,j') + d(j',j'') \geqslant d(j,j'')$, removing jobs planned twice decrease the objective function, there is an optimal solution allocating exactly one technician per job. Constraints (5) express that the depots d_i are the flow source and arrival for all technician i. Constraints (6), (7) and (8) express time windows constraints, respectively between two consecutive jobs, and the respect of the technicians' starting and finishing time. Coefficients $M^1_{j,j'}$, $M^2_{i,j}$, $M^3_{i,j}$ can be chosen as following: $M^1_{j,j'} = \tilde{t}^{max}_j + D_j + T(j,j') - \tilde{t}^{min}_{j'}$, $M^2_{i,j} = \tilde{t}^{start}_i + T(i,j) - \tilde{t}^{min}_j$ and $M^3_{i,j} = \tilde{t}^{max}_j + D_j + T(j,i) - \tilde{t}^{end}_i$.

To increase the solving limits of this weak MIP formulation, matheuristics are efficient to tackle larger instances, even for highly-constrained instances, as shown in [7]. MIP neighborhoods restrict the number of technician and the number of jobs in the MIP computations, for greedy algorithms constructing solutions from scratch, or for Variable Neighborhood Descent (VND) strategies.

3 Extended formulation and CG scheme

Similarly to [9], we have an extended MIP formulation by enumerating all possible (and non empty) routes $\mathcal{P}_i$ for each technician i. The cost of a route $k \in \mathcal{P}_i$ is denoted $C_{i,k}$. The variables of the MIP formulation are $z_{i,k} \in \{0,1\}$, such that $z_{i,k} = 1$ if route $k \in \mathcal{P}_i$ is chosen, and $y_j \in \{0,1\}$ with $y_j = 1$ if job j is not

planned. It leads to the following MIP:

$$\min_{y,z} \sum_{i\in\mathcal{I},k\in\mathcal{P}_i} C_{i,k}z_{i,k} + \sum_{j\in\mathcal{J}} P_j y_j \tag{10}$$

$$s.t: \quad \sum_{i,k:j\in k} z_{i,k} + y_j \geqslant 1 \qquad \forall j \in \mathcal{J}, \tag{11}$$

$$\sum_{k\in\mathcal{P}_i} z_{i,k} \leqslant 1 \qquad \forall i \in \mathcal{I}, \tag{12}$$

$$z_{i,k}, y_j \in \{0,1\} \tag{13}$$

The difficulty is that $\mathcal{P}_i$ cannot be enumerated. The Linear Programming (LP) relaxation is solved by the CG algorithm, adding iteratively new routes. Having a subset of routes $\mathcal{C}_i \subset \mathcal{P}_i$, the Restricted Master Problem (RMP) denotes the previous LP relaxation applied to a subset of variables:

$$\begin{array}{ll} \min\limits_{z_{i,k},y_j\geqslant 0} & \sum_{i\in\mathcal{I},k\in\mathcal{C}_i} C_{i,k}z_{i,k} + \sum_{j\in\mathcal{J}} P_j y_j \\ s.t: & \sum_{i,k:j\in k} z_{i,k} + y_j \geqslant 1 \qquad (\pi)\ \forall j \in \mathcal{J}, \\ & \sum_{k\in\mathcal{C}_i} z_{i,k} \leqslant 1 \qquad (\sigma)\ \ \forall i \in \mathcal{I}, \end{array} \tag{14}$$

π, σ denotes respectively the dual variables associated to the constraints (11),(12).

Variables must be added in the RMP if their reduced cost is strictly negative. If the remaining variables have a positive reduced cost, the RMP gives the LP relaxation of (10-12). This negativity question among a non enumerable set is formulated as an optimization problem, minimizing the reduced costs of the columns that are not in $\cup_i \mathcal{C}_i$. These negativity question are decomposed for all technicians, $RC^* = \min_i RC_i^*$, where RC_i^* is the minimization problem over the possible routes for technician i, $RC^* \geqslant 0$ ensures that the RMP gives the LP relaxation of (10-12). Computations of RC_i^* are independent for the technicians, and can be formulated similarly to the MIP formulation of Section 2:

$$\begin{array}{ll} RC_i^* = & \min_u \sigma_i + \sum_{j,j'} (D(j,j') - \pi_j) u_{j,j'} \\ s.t: \forall j \in \mathcal{J}, & \sum_{j'\in\mathcal{J}\cup\{d_i\}} u_{j'j} = \sum_{j'\in\mathcal{J}\cup\{d_i\}} u_{jj'} \\ & \sum_{j'\in I} u^j_{j'd_i} = \sum_{j'\in I} u^j_{d_ij'} \leqslant 1 \\ \forall (j,j') \in \mathcal{J}, & t_j + D_j + T(j,j') \leqslant t_{j'} + (1 - u_{j,j'}) \cdot M^3_{i,j} \\ j \in \mathcal{J}, & \tilde{t}_i^{start} + T(i,j) \leqslant t_j + (1 - u_{d_ij}) \cdot M^2_{i,j} \\ j \in \mathcal{J}, & t_j + D_j + T(j,i) \leqslant \tilde{t}_i^{end} + (1 - u_{jd_i}) \cdot M^1_{j,j'} \\ \forall j,j' & u_{j'j} \in \{0,1\}, t_j \in [\tilde{t}_j^{min}, \tilde{t}_j^{max}] \end{array} \tag{15}$$

The LP relaxation of (10-12) is calculated by the Algorithm 1, with an initial set of columns as input. The initial set of columns can be empty, in this case the first RMP is feasible thanks to the penalty costs. Matheuristics can initialize the CG algorithm adding the columns corresponding to primal solutions.

To solve CG subproblems (15) to optimality, a dynamic programming algorithm is valid formulating (15) as a shortest path problem with resource constraints (ESPPRC), we refer to [10]. However, the computations to optimality are required only to prove the optimality of the RMP at the last iteration. For the first CG iterations, we just need to generate negative reduced cost solutions. The dynamic programming computations are time consuming in the Algorithm

Algorithm 1: Standard column generation algorithm

Input:
  $\mathcal{C}$ set of initial columns.
do :
  solve RMP (14) with columns defined in $\mathcal{C}$
  store dual variables σ and π and optimal cost from (14)
  for each technician $i \in \mathcal{I}$:
    solve (15) to optimality with last (σ, π) values
    if $CR_i^* < 0$ **then** add the optimal column to $\mathcal{C}$
  end for
while : columns are added in $\mathcal{C}$
return the last cost of the RMP (14)

1, it can be replaced by heuristics to generate quickly negative solutions. For instance, the matheuristics [7] apply.

A general difficulty to implement CG algorithms is that the dual variables converge erratically. This is closely related to the properties of linear optimization as the optimums are extreme points. A consequence for the CG algorithm is that many iterations are necessary to have good dual variables to generate the most appropriate columns. Stabilization techniques enforce to smooth the convergence of dual variables to reduce the number of iterations to converge for the CG algorithm, and thus to reduce the total computation time for the CG convergence. To stabilize the convergence of dual variables and to reduce the computational effort, stabilization techniques are essential. Mathematical properties can be exploited as in [2, 4, 18], but also heuristics as in [12]. In the following, tailored stabilization schemes are proposed using matheuristics.

Algorithm 1 computes the LP relaxation of (10-12). Such iterations, even truncated before the stopping criterion, can also be used to design heuristics. Integer computations of the RMP with the generated columns give primal solutions, which is efficient in the computational experiments of [7]. Similarly to Lagrangian heuristics, the partial LP relaxations can be repaired to build primal solutions. The effort to accelerate the convergence of the Algorithm 1 has also an impact to accelerate CG heuristics.

4 POPMUSIC column generation stabilization

This section is motivated by the following fact: dual variables induce to generate columns with jobs with the highest values π_j in (15). Hence, independent computations of (15) are likely to use the same jobs, with the highest values of π_j. These generated columns are likely to be redundant in the recombination induced by the next RMP computation. It suggest to incorporate diversifications in the generation of columns. To achieve that goal, we investigate combined subproblems for a subset of technicians $\mathcal{I}_0 \subset \mathcal{I}$ (typically two or three technicians), imposing a total diversification in these subproblems, with following MIP subproblems:

$$
\begin{array}{lll}
RC^*_{\mathcal{I}_0} = & \max_u \sum_{i\in\mathcal{I}_0} C_i & (16.1)\\
s.t : \forall j \in \mathcal{J} & \sum_{i\in\mathcal{I}_0, j'\in\mathcal{J}} (u^i_{jj'} + u^i_{jd_i}) \leqslant 1 & (16.2)\\
\forall i \in \mathcal{I}_0, & -C_i \leqslant +\sigma_i + \sum_{j,j'} (D(j,j') - \pi_j) u^i_{j,j'} & (16.3)\\
\forall i \in \mathcal{I}_0, \forall j \in \mathcal{J}, & \sum_{j'\in\mathcal{J}\cup\{d_i\}} u^i_{j'j} = \sum_{j'\in\mathcal{J}\cup\{d_i\}} u^i_{jj'} & (16.4)\\
\forall i \in \mathcal{I}_0, & \sum_{j'\in I} u^i_{j'd_i} = \sum_{j'\in I} u^i_{d_ij'} \leqslant 1 & (16.5)\\
\forall i \in \mathcal{I}_0, \forall (j,j') \in \mathcal{J}, & t_j + D_j + T(j,j') \leqslant t_{j'} + \left(1 - u^i_{j,j'}\right) \cdot M^3_{i,j} & (16.6)\\
\forall i \in \mathcal{I}_0, j \in \mathcal{J}, & \tilde{t}^{start}_i + T(i,j) \leqslant t_j + \left(1 - u^i_{d_ij}\right) \cdot M^2_{i,j} & (16.7)\\
\forall i \in \mathcal{I}_0, j \in \mathcal{J}, & t_j + D_j + T(j,i) \leqslant \tilde{t}^{end}_i + \left(1 - u^i_{jd_i}\right) \cdot M^1_{j,j'} & (16.8)\\
\forall i \in \mathcal{I}_0, & C_i \geqslant 0 & (16.9)\\
\forall j, j' & u^i_{j'j} \in \{0,1\}, t_j \in [\tilde{t}^{min}_j, \tilde{t}^{max}_j] &
\end{array}
\qquad (16)
$$

This subproblem can be seen as a concatenation of the technician subproblems in $\mathcal{I}_0 \subset \mathcal{I}$, with (16.2) to enforce that each job can be affected to at least one technician in $\mathcal{I}_0$. Furthermore constraints (16.3) and objective function (16.1) induce an objective function where a technician without negative reduced cost counts for 0, and the global optimization minimizes the sum of the negative reduced cost among the technicians. It gives rise to the Algorithm 2, where the partitions of technicians are varying for each iterations, similarly to POPMUSIC (Partial optimization metaheuristic under special intensification conditions [20]).

Algorithm 2: POPMUSIC column generation algorithm

Input:
  $\mathcal{C}$ set of initial columns.
do :
  solve RMP (14) with columns defined in $\mathcal{C}$
  store dual variables σ and π and optimal cost from (14)
  compute $\mathcal{P}_I$ a partition of $\mathcal{I}$ in small subsets
  for each subset $\mathcal{I}_0 \in \mathcal{P}_I$:
    solve (16) with a matheuristic with last (σ, π) values
    for each column c with a negative reduced cost
      add the column to $\mathcal{C}$
    end for
  end for
while : columns are added in $\mathcal{C}$
return the last cost of the RMP (14)

The subproblems (16) can be solved to search columns with a negative reduced cost using matheuristics, the number of technician is restricted and matheuristics restrict the size of the MIP computations removing jobs similarly to [7]. With several technicians in the subproblems (16), the dynamic programming algorithm [10] does not apply anymore. The POPMUSIC CG scheme applies only to generate columns with a negative reduced cost. To prove the optimality of the RMP, Algorithm 1 applies for the remaining iterations.

Algorithm 3: Diversification of subproblem solutions

Input:
- $\mathcal{I}_0$ a subset of technician.
- s a cyclic permutation of $\mathcal{I}_0$ with order$(s) = |\mathcal{I}_0|$.
- σ, π the dual variables of the last RMP computation.

Initialization: $\mathcal{C} = \varnothing$, the columns to add in the RMP
for each technician $i \in \mathcal{I}_0$:
  Let $i' = i_0$, $\mathcal{J}_0 = \mathcal{J}$
  for $k = 1$ to $|\mathcal{I}_0|$:
    solve (15) for technician i' with (σ, π) values and the remaining jobs in $\mathcal{J}_0$
    if the solution induces a column with a negative reduced cost
      add the column in $\mathcal{C}$
      remove the jobs of the column in $\mathcal{J}_0$
    $i' = s(i_0)$
  end for :
end for
return $\mathcal{C}$

Algorithm 3 decomposes the search of columns with a negative reduced cost using only computations with one technician. These decomposed subproblems can also be solved as ESPPRC with [10]. This scheme allows to generate the best individual columns as in Algorithm 1, and also good complementary columns for these best individual columns. The solutions of (15) with Algorithm 3 are uneven in the reduced costs following the order of priority of the technicians, whereas a straightforward matheuristic search applied to (15) would have a smoother repartition of the reduced costs. A stake of the computational experiments is to determine whether these two generation schemes are complementary or if one dominates the other. To generate both types of columns, Algorithm 3 can be applied to generate initial columns before a smoothing local search phase to optimize the cost (16.1) with a MIP-VND similar to [7].

5 Tabu Search Matheuristic intensification

This section is motivated by the following fact: many good columns can be obtained with slight modifications from a good column. At one iteration, high values of π indicate the jobs that are highly encouraged to be generated in a new column. A kernel of jobs with high π_j values is likely to be combined in several interesting columns regarding the reduced costs. Using such property, it is interesting to generate aggressively columns with local search moves from interesting columns. This section introduces a Tabu Search (TS) matheuristic to generate quickly a pool of good solutions.

We denote the binaries $v_{i,j} = \sum_{j'} u^i_{jj'}$ indicating if technician $i \in \mathcal{I}_0$ realizes job j. Having N feasible solutions previously calculated, we denote with $\tilde{v}^N_{i,j}$ the value of these variables. To forbid already generated columns, we add the

following "no-good-cuts" constraints similarly to [14]:

$$\forall n \in [\![1, N]\!], \sum_{i,j:\tilde{v}^n_{i,j}=1} (1 - v^n_{i,j}) + \sum_{i,j:\tilde{v}^n_{i,j}=0} v^n_{i,j} \geqslant 1 \tag{17}$$

To search around the last solution $\tilde{v}^N_{i,j}$, allowing k modifications from the N-th solution, it can also be written as a linear constraints:

$$\sum_{i,j:\tilde{v}^N_{i,j}=1} (1 - v^N_{i,j}) + \sum_{i,j:\tilde{v}^N_{i,j}=0} v^N_{i,j} \leqslant k \tag{18}$$

Algorithm 4: Tabu search intensification

Input:

- $\mathcal{I}_0$ a subset of technicians
- the current value in the RMP of dual variables (σ, π)
- a set of initial columns $c \in \prod_{i \in \mathcal{I}_0} \mathcal{P}_i$ with a negative reduced cost in (15)
- an integer $N \in \mathbb{N}$, a maximal number of TS iterations
- an integer $k \in \mathbb{N}$, a number of maximal modifications

TSintensification$(\mathcal{I}_0, k, N)$
//Initialization:
MIP, a MIP formulation for (15) related to technician i
$p = c$ initial columns
Taboo list of columns $l = \{c\}$
an integer $n = 0$ to denote iterations
//Loop to generate columns with negative reduced costs
do:
Add constraint (18) in MIP with columns of p
Add constraint (17) in MIP with columns of p
solve MIP
remove constraint (18) in MIP with columns of p
update p, the optimal columns in the last MIP
update l adding the columns of p with a negative reduced cost in l
while $n < N$ and ReducedCost$(p) < 0$
return l // the list of column to add in the next RMP

These constraints allow to generate a TS procedure in Algorithm 4 to generate aggressively columns for the classic CG or the last POPMUSIC CG scheme. In the case of the classic CG algorithm, the input for $\mathcal{I}_0$ is a singleton in the Algorithm 4. Adding such constraints in the CG subproblems can be solved with Branch&Bound for small MIP computations. For larger MIP computations, these constraints respect the structures of the MIP matheuristics from [6] to find quickly good solutions.

6 General algorithm

Algorithm 5 gathers POPMUSIC and TS stabilization of the CG algorithm, both techniques can be processed together. With parameter $N = -1$, it deactivates TS stabilization. To deactivate POPMUSIC stabilization, the partitioning procedure always split $\mathcal{I}$ into singletons.

Algorithm 5: Stabilized column generation with matheuristics

Input:
$N \in \mathbb{N}, k \in \mathbb{N}$, TS parameters, $N = -1$ deactivates tabu search;
a partitioning procedure among technicians;
p a pool of matheuristics to solve subproblems (16);
a stopping criterion for the final CG;
$\mathcal{C}$ = set of initial columns.
do :
solve RMP (14) with columns defined in $\mathcal{C}$
store dual variables σ and π and optimal cost from (14)
Compute $\mathcal{P}_I$ a partition of $\mathcal{I}$ in small subsets.
for each subset $\mathcal{I}_0 \in \mathcal{P}_I$:
solve (16) with a matheuristic with last (σ, π) values
if $N > 0$ **then** apply TSINTENSIFICATION$(\mathcal{I}_0, k, N)$
for each column c with a negative reduced cost
add the column to $\mathcal{C}$
end for
end for
while : columns are added in $\mathcal{C}$
`// Final column generation phase`
do :
solve RMP (14) with columns defined in $\mathcal{C}$
store dual variables σ and π and optimal cost from (14)
for each technician $i \in \mathcal{I}$:
solve (15) to optimality with last (σ, π) values
if $N > 0$ **then** apply TSINTENSIFICATION$(\{i\}, k, N)$
add in $\mathcal{C}$ the columns with a negative reduced cost
end for
while : columns are added in $\mathcal{C}$ or the specific stopping criterion is reached
return the current RMP solution and the last cost of the RMP (14)

7 Computational results

The computational experiments were computed with a laptop Intel Core2 Duo processor, 2.80GHz, running Linux Ubuntu. The code was implemented in Python using PuLP to call MIP solvers like Cplex or open source tools with the same programming interface. The instances for the computational experiments are

the ones from [7], with a graduation of highly-constrained instances to instances more similar to VRPTW instances with no skill constraints.

Algorithm 5 provides a very significant acceleration of the CG iterations for the CG heuristics of [7]. Solving subproblems with matheuristics provides firstly a significant acceleration. We note that the quality of MIP-based matheuristics to solve the subproblems (15) allows to have fewer iterations with the exact computations at the end of Algorithm 5. To run repairing heuristics based on the RMP, two iterations were enough for the last phase of Algorithm 5.

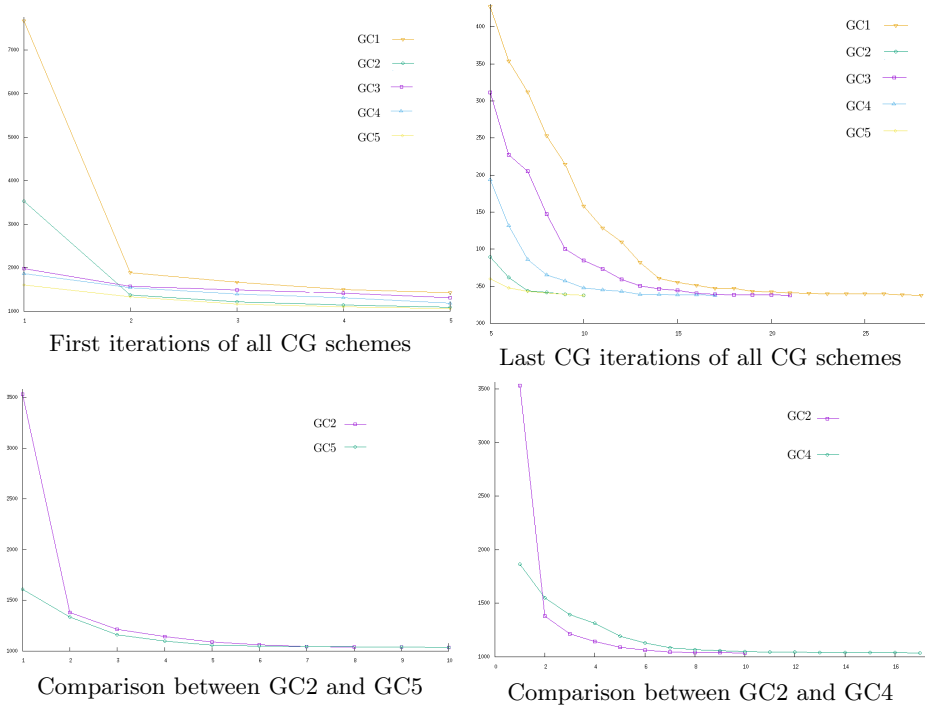

Fig. 1. Comparison of the CG convergence for an instance with 40 jobs and 15 technicians with 10 different skills

The acceleration of the CG algorithm concerned also the reduction of the number of iterations to converge. The aggressive generation of several columns for each subproblems with several matheuristics is a first way to stabilize the CG algorithm. To understand the specific contributions of the different CG schemes presented in this paper, we provide here comparative analyses based on several schemes solving exactly subproblems for small and mid-size instances, and a proven termination of the CG algorithm. Following CG schemes are implemented and compared:

– `CG1`: it is the classic CG scheme, as written in the Algorithm 1.

- `CG2`: The classical CG scheme is deployed with a TS intensification to solve single technician subproblems. It corresponds to the Algorithm 5 with only partitions into singletons, with parameters $k = 3$ and $N = 5$.
- `CG3`: it is the POPMUSIC CG scheme without TS intensification, where the subproblems (16) are solved only using the Algorithm 3 and exact computations.
- `CG4`: it is the POPMUSIC CG scheme without TS intensification, where the subproblems (16) are solved twice: Algorithm 3 gives first solutions and a second phase optimize the summed reduced cost (and thus balancing the reduced costs) with a VND similar to [7], generating all the columns with a negative reduced cost got from these two phases.
- `CG5`: it corresponds to the strategy `CG4` with TS intensification activated with $k = 3$ and $N = 5$.

The dominant computational results are illustrated in Fig.1. The known hierarchies among CG schemes are significant: `CG1` is significantly dominated by the other schemes, POPMUSIC and TS intensification induce a very significant acceleration of the CG convergence. Comparing `CG3` and `CG4` allows to conclude that the optimization of the summed reduced cost in subproblems (16) induces a better CG convergence than the CG with the hierarchical decomposition of Algorithm 3. Combining POPMUSIC and TS intensification dominates the single stabilization strategies: POPMUSIC column generation is mostly useful for the first iterations to recombine columns in the next RMP, where it can be difficult to find a solution allocating all the jobs (and thus paying no prohibitive penalty). TS intensification is crucial when dual variables have a relative stability, which is helpful for the latest phases of the CG convergence. To compute the extended LP relaxation to optimality, TS strategies are prominent to reduce the number of CG iterations.

8 Conclusion and perspectives

This work outlined how matheuristics can be used to solve different subproblems from the classical CG schemes to reduce the time required for the convergence of a CG scheme. It induces a better stability of the dual variables and reduces the number of iterations required for the CG convergence. More specifically, POPMUSIC CG imposes a diversification of columns which is useful in the first stages, whereas an aggressive generation of solution with a Tabu Search matheuristic is efficient when dual are stabilized. Combining both techniques showed promising results.

The perspectives of these results are to decrease the time necessary for a CG convergence. It increases the size of instances where dual bounds are computable. It accelerated also CG matheuristics having earlier good dual variables to generate columns for primal heuristics. As a perspective, the CG strategies proposed in this work can be applied to other problems where CG induce independent and heterogeneous subproblems. The comparison/combination of the

matheuristic stabilization with exact stabilization methods will also be investigated.

References

1. A. Agra et al. The robust vehicle routing problem with time windows. *Computers&OR*, 40(3):856–866, 2013.
2. H. Amor, J. Desrosiers, and A. Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009.
3. C. Archetti and M.G. Speranza. A survey on matheuristics for routing problems. *EURO J on Comput Optim*, 2(4):223–246, 2014.
4. O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 94:229–237, 1999.
5. N. Dupin and E. Talbi. Dual Heuristics and New Lower Bounds for the Challenge EURO/ROADEF 2010. *Matheuristics 2016*, pages 60–71, 2016.
6. N. Dupin and E. Talbi. Matheuristics for the Discrete Unit Commitment Problem with Min-Stop Ramping Constraints. *Matheuristics 2016*, pages 72–83, 2016.
7. N. Dupin and E. Talbi. Matheuristics for a VRPTW with competence constraints. In *12th Metaheuristics International Conference MIC2017*, 2017.
8. P. Dutot, A. Laugier, and A. Bustos. Technicians and interventions scheduling for telecommunications. *France Telecom R&D*, 2006.
9. D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR: A Quarterly Journal of Operations Research*, pages 407–424, 2010.
10. D. Feillet et al. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.
11. M. Fırat and C. Hurkens. An improved MIP-based approach for a multi-skill workforce scheduling problem. *Journal of Scheduling*, 15(3):363–380, 2012.
12. P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.
13. A. Kovacs et al. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of scheduling*, 15(5):579–600, 2012.
14. J. Lazic et al. Variable Neighbourhood Decomposition Search for 0-1 Mixed Integer Programs. *Computers& OR*, 37(6):1055–1067, 2010.
15. M. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
16. V. Pillac, C. Gueret, and A. Medaglia. A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, pages 1–11, 2013.
17. S. Pokutta and G. Stauffer. France telecom workforce scheduling problem: a challenge. *RAIRO-Operations Research*, 43(4):375–386, 2009.
18. L-M Rousseau, M. Gendreau, and D. Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668, 2007.
19. E. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, 33(1):1–14, 1999.
20. É. Taillard and S. Voss. POPMUSIC-Partial optimization metaheuristic under special intensification conditions. In *Essays and surveys in metaheuristics*, pages 613–629. Springer, 2002.